What is claimed is:

1. A method for implementing class of service among a plurality of clients sending requests seeking access to sites hosted on a plurality of back-end servers, comprising the steps of:

grouping at least one of said plurality of back-end servers into a respective one of a plurality of service classes;

receiving a client request for host access at a front end processor;

selecting a class of service from said plurality of service classes according to at least one selected parameter of said request; and

distributing said request to a back-end server in said selected class of service according to the load of each of said at least one of said back-end servers in the selected service class.

2. The method of claim 1 in which said selected parameters of the request are selected from a group consisting of: user authentication, virtual site level class of service and client level class of service;

wherein a user authentication identifies a subscribed class for an authenticated user;

a virtual site level class of service is determined by host name and selected protocol; and

a client level class of service is determined as a function of the request/transaction, service/protocol, authenticated user, URL, destination port, domain of origin, source IP, destination IP, and application requested.

3. The method of claim 1 in which said step of distributing the request according to the load further includes a load balancing algorithm selected from the group consisting of: weighted percentage; round robin; CPU availability; least connections; and probabilistic.

4. A method for implementing probabilistic load balancing among a plurality of back-end servers, comprising the steps of:

determining the load on each of said plurality of back-end servers, including a maximum load L;

calculating the difference between the maximum load L + 1 and the load of each of said plurality of back-end servers to obtain a respective value for each back-end server;

summing said respective values to obtain a value D; and

proportionally distributing the next D requests to the plurality of back-end servers based on their respective values such that more requests are routed to servers having a relatively light load.

5. A method for implementing a rules based adaptive policy engine for real-time balancing of incoming requests across a plurality of back-end servers comprising the steps of:

clustering said plurality of back-end servers according to class of service, each cluster having a corresponding class of service;

receiving a request characterized as belonging to a designated class of service;

checking to see if said request is a new session or an old session, where said old session has an identified back-end server and contents;

if said request is an old session, then

checking to see if said identified back-end server and contents are available;

if said request is a new session, or if said identified back-end server and content are not available,

running a load balancing algorithm selected for the designated class of service; and

routing said request to a back-end server within said cluster having the designated class of service, as selected by the load balancing algorithm.

6. The method of claim 5 in which said load balancing algorithm is selected from the group consisting of: weighted percentage; round robin; CPU availability; least connections; and probabilistic.

7. The method of claim 5 in which the step of running a load balancing algorithm further comprises the steps of:

deploying an intelligent agent on at least one of said plurality of back-end servers;

using said intelligent agent to collect information in the form of selected server attributes;

transmitting said collected information to an adaptive policy engine; and

selecting a back-end server according to the collected server attributes.

8. The method of claim 7 in which said step of transmitting said collected information further includes:

a network transmission step of multicasting a UDP packet containing information related to said selected server attributes; and

a network reception step of decoding the UDP packet in the adaptive policy engine.

9. A method for implementing collection of information for monitoring the performance in a cluster of web servers, comprising the steps of:

providing an adaptive policy engine;

deploying an intelligent agent on at least one of said plurality of web servers;

using said intelligent agent to collect information in the form of selected server attributes;

transmitting said collected information to said adaptive policy engine for use in dynamically allocate requests to selected web servers in said cluster of web servers to meet pre-defined Service Level Agreements (SLA).

10. The method of claim 9 further including the step of repackaging at least some of said collected information into a management information base.

11. The method of claim 9 further including the step of making at least some of said collected information available to an application program for monitoring real-time web server performance.

5